```
 1: // $Id: extern.h,v 1.1 2018-04-02 15:31:00-07 - - $
 2:
 3: #ifndef __EXTERN_H__
 4: #define __EXTERN_H__
 5:
 6: int yylex();
 7: int yyparse();
 8:
 9: void badchar (char);
10: void yyerror (const char*);
11:
12: extern char* yytext;
13:
14: #define YYSTYPE double
15: #include "yyparse.h"
16:
17: #endif
18:
```

```
 1: /* $Id: lexer.l,v 1.2 2018-09-26 16:37:28-07 - - $ */
 2:
 3: %{
 4:
 5: #include "extern.h"
 6:
 7: %}
 8:
 9: %option noyywrap
10:
11: MANTISSA    ([[:digit:]]+\.?[[:digit:]]*|\.[[:digit:]]+)
12: EXPONENT    ([Ee][+-]?[[:digit:]]+)
13: INFINITY    ([Ii][Nn][Ff][[:isalnum:]]*)
14: NAN         ([Nn][Aa][Nn][[:isalnum:]]*)
15: NUMBER      ({MANTISSA})({EXPONENT})?|INFINITY|NAN
16:
17:
18: %%
19:
20: "#".*       { ECHO; }
21: [ \t]+      { ECHO; }
22: \n          { ECHO; return '\n'; }
23: {NUMBER}    { ECHO; return NUMBER; }
24: "+"         { ECHO; return '+'; }
25: "-"         { ECHO; return '-'; }
26: "*"         { ECHO; return '*'; }
27: "/"         { ECHO; return '/'; }
28: .           { ECHO; printf ("\ninvalid character \\0%o\n", *yytext); }
29:
30: %%
```

```
 1: // $Id: parser.y,v 1.3 2018-09-26 16:42:13-07 - - $
 2:
 3: %{
 4:
 5: #include <stdio.h>
 6: #include <stdlib.h>
 7:
 8: #include "extern.h"
 9:
10: %}
11:
12: %token NUMBER
13:
14: %%
15:
16: input : input expr '\n' { printf ("\nanswer: %.15g\n", $2); }
17:       | '\n'           { printf ("\n"); }
18:       | error '\n'     { printf ("\n"); }
19:       |
20:       ;
21:
22: expr  : NUMBER         { $$ = atof (yytext); }
23:       | expr expr '+'  { $$ = $1 + $2; }
24:       | expr expr '-'  { $$ = $1 - $2; }
25:       | expr expr '*'  { $$ = $1 * $2; }
26:       | expr expr '/'  { $$ = $1 / $2; }
27:       ;
28:
29: %%
30:
```

```
 1: // $Id: main.cpp,v 1.2 2018-09-26 16:37:28-07 - - $
 2:
 3: #include <stdio.h>
 4: #include <stdlib.h>
 5:
 6: #include "extern.h"
 7:
 8: void yyerror (const char* error) {
 9:    printf ("%s\n", error);
10: }
11:
12: int main () {
13:    yyparse();
14:    return EXIT_SUCCESS;
15: }
```

```
 1: # $Id: Makefile,v 1.7 2019-04-01 18:28:26-07 - - $
 2:
 3: WARN   = -Wall -Wextra -Werror -Wpedantic -Wshadow -Wold-style-cast
 4: CPP    = g++ -g -O0 -std=gnu++17 ${WARN} -fdiagnostics-color=never
 5: CPPYY  = g++ -g -O0 -std=gnu++17 -Wno-register -fdiagnostics-color=never
 6: OBJS   = main.o yyparse.o yylex.o
 7: GENS   = yyparse.h yyparse.cpp yylex.cpp
 8: SRCS   = extern.h lexer.l parser.y main.cpp Makefile
 9:
10: all : rpn
11:
12: rpn : ${OBJS}
13:         ${CPP} -o rpn ${OBJS}
14:
15: %.o : %.cpp
16:         checksource $<
17:         cpplint.py.perl $<
18:         ${CPP} -c $<
19:
20: yyparse.cpp yyparse.h : parser.y
21:         bison --defines=yyparse.h --output=yyparse.cpp parser.y
22:
23: yyparse.o : yyparse.cpp
24:         ${CPPYY} -c yyparse.cpp
25:
26: yylex.cpp : yyparse.h lexer.l
27:         flex --outfile=yylex.cpp lexer.l
28:
29: yylex.o : yylex.cpp
30:         ${CPPYY} -Wno-register -c yylex.cpp
31:
32: spotless : clean
33:         - rm rpn Makefile.deps Listing.{ps,pdf}
34:
35: clean :
36:         - rm ${OBJS} ${GENS} test.out
37:
38: ci :
39:         - cid + ${SRCS}
40:
41: test : rpn
42:         ( echo "3 4 * 1 3 / +" \
43:         ; echo "10 0 /" \
44:         ; echo foo \
45:         ) | tee test.in | rpn >test.out 2>&1
46:         more test.in test.out >test.listing </dev/null
47:         - rem test.in test.out
48:
49: lis : ${SRCS} test
50:         mkpspdf Listing.ps ${SRCS} yyparse.h test.listing
51:
52: again : spotless
53:         ${MAKE} ci all lis
54:
55: Makefile.deps : yyparse.cpp yylex.cpp
56:         ${CPP} -MM yyparse.cpp yylex.cpp main.cpp >Makefile.deps
57:
58: include Makefile.deps
```

```
59:
```

```
    1: /* A Bison parser, made by GNU Bison 3.0.4.  */
    2:
    3: /* Bison interface for Yacc-like parsers in C
    4:
    5:    Copyright (C) 1984, 1989-1990, 2000-2015 Free Software Foundation, In
c.
    6:
    7:    This program is free software: you can redistribute it and/or modify
    8:    it under the terms of the GNU General Public License as published by
    9:    the Free Software Foundation, either version 3 of the License, or
   10:    (at your option) any later version.
   11:
   12:    This program is distributed in the hope that it will be useful,
   13:    but WITHOUT ANY WARRANTY; without even the implied warranty of
   14:    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   15:    GNU General Public License for more details.
   16:
   17:    You should have received a copy of the GNU General Public License
   18:    along with this program.  If not, see <http://www.gnu.org/licenses/>.
  */
   19:
   20: /* As a special exception, you may create a larger work that contains
   21:    part or all of the Bison parser skeleton and distribute that work
   22:    under terms of your choice, so long as that work isn't itself a
   23:    parser generator using the skeleton or a modified version thereof
   24:    as a parser skeleton.  Alternatively, if you modify or redistribute
   25:    the parser skeleton itself, you may (at your option) remove this
   26:    special exception, which will cause the skeleton and the resulting
   27:    Bison output files to be licensed under the GNU General Public
   28:    License without this special exception.
   29:
   30:    This special exception was added by the Free Software Foundation in
   31:    version 2.2 of Bison.  */
   32:
   33: #ifndef YY_YY_YYPARSE_H_INCLUDED
   34: # define YY_YY_YYPARSE_H_INCLUDED
   35: /* Debug traces.  */
   36: #ifndef YYDEBUG
   37: # define YYDEBUG 0
   38: #endif
   39: #if YYDEBUG
   40: extern int yydebug;
   41: #endif
   42:
   43: /* Token type.  */
   44: #ifndef YYTOKENTYPE
   45: # define YYTOKENTYPE
   46:   enum yytokentype
   47:   {
   48:     NUMBER = 258
   49:   };
   50: #endif
   51:
   52: /* Value type.  */
   53: #if ! defined YYSTYPE && ! defined YYSTYPE_IS_DECLARED
   54: typedef int YYSTYPE;
   55: # define YYSTYPE_IS_TRIVIAL 1
   56: # define YYSTYPE_IS_DECLARED 1
```

```
57: #endif
58:
59:
60: extern YYSTYPE yylval;
61:
62: int yyparse (void);
63:
64: #endif /* !YY_YY_YYPARSE_H_INCLUDED  */
```

```
  1: ::::::::::::::
  2: test.in
  3: ::::::::::::::
  4: 3 4 * 1 3 / +
  5: 10 0 /
  6: foo
  7: ::::::::::::::
  8: test.out
  9: ::::::::::::::
 10: 3 4 * 1 3 / +
 11:
 12: answer: 12.3333333333333
 13: 10 0 /
 14:
 15: answer: inf
 16: f
 17: invalid character \0146
 18: o
 19: invalid character \0157
 20: o
 21: invalid character \0157
 22:
 23: syntax error
 24:
```